

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

MODELOVÁNÍ A REALIZACE SYSTÉMŮ PRO INTELIGENTNÍ BUDOVY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ZUZANA KONEČNÁ

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

MODELOVÁNÍ A REALIZACE SYSTÉMŮ PRO INTELIGENTNÍ BUDOVY

MODELLING AND REALIZATION OF SYSTEMS FOR INTELLIGENT HOUSES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ZUZANA KONEČNÁ

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. FRANTIŠEK ZBOŘIL, Ph.D.

BRNO 2015

Abstrakt

Cílem této bakalářské práce bylo seznámení s PLC systémem Foxtrot firmy Tecomat. Následně bylo úkolem navrhnout pro tento systém ovládání rodinného domu s propojením z bezdrátovou senzorovou sítí. Jako rozhraní pro toto spojení byl použit počítač Raspberry Pi

Abstract

The aim of this thesis was to get acquainted with the PLC system Foxtrot fom Teco a.s. company. Subsequently, the task was to design a system for the control of the house with a connection with a wireless sensor network. Raspberry Pi was used as an interface for this connection

Klíčová slova

inteligentní elektroinstalace, Foxtrot Tecomat, bezdrátová senzorová síť, Raspberry Pi.

Keywords

Intelligent wiring, Foxtrot Tecomat, wireless sensor network, Raspberry Pi.

Citace

Zuzana Konečná: Modelování a realizace systémů pro inteligentní budovy, bakalářská práce, Brno, FIT VUT v Brně, 2015

Modelování a realizace systémů pro inteligentní budovy

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana doc.
Ing. Františka Zbořila Ph.D

.....
Zuzana Konečná
20. května 2015

Poděkování

Ráda bych poděkovala panu Jakubu Žákovi, Ing. za odbournou pomoc s bezdrátovou sen-
zorovou sítí.

© Zuzana Konečná, 2015.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informa-
čních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění
autorem je nezákonné, s výjimkou zákonem definovaných případů.*

Obsah

| | |
|---|-----------|
| 1 Úvod | 3 |
| 2 Analýza systémů | 4 |
| 2.1 Inteligentní dům | 4 |
| 2.1.1 Historie | 4 |
| 2.2 Co je to inteligentní dům ? | 4 |
| 2.3 V současnosti používané systémy | 5 |
| 2.4 Základní principy inteligentní elektroinstalace | 6 |
| 2.4.1 Prvky inteligentního systému | 6 |
| 2.4.2 Rozdělení systémů | 6 |
| 2.5 Tecomat Foxtrot | 8 |
| 2.5.1 Foxtrot CP-1000 | 9 |
| 2.5.2 Norma IEC 61 131 | 11 |
| 2.5.3 Vývojové prostředí Mosaic | 12 |
| 2.5.4 Prvotní nastavení | 15 |
| 2.6 Bezdrátové senzorové sítě | 17 |
| 2.6.1 Základní principy bezdrátových senzorových sítí | 17 |
| 2.6.2 Platforma JADE | 17 |
| 2.7 Raspberry Pi | 19 |
| 3 Návrh realizace | 20 |
| 3.1 Komponenty | 20 |
| 3.2 Implementace | 22 |
| 3.2.1 Příklad funkčního bloku | 22 |
| 3.2.2 Topení | 23 |
| 3.2.3 Žaluzie | 23 |
| 3.2.4 Rekuperace | 24 |
| 3.2.5 Komplexní regulace teploty | 25 |
| 3.2.6 Zabezpečení | 26 |
| 3.3 Propojení systému se senzorovou sítí | 27 |
| 3.4 Ovládání | 28 |
| 4 Závěr | 30 |
| A Obsah CD | 33 |

Seznam obrázků

| | | |
|------|---|----|
| 2.1 | Decentralizovaný systém [2] | 7 |
| 2.2 | Centralizovaný systém [2] | 7 |
| 2.3 | Hybridní systém [2] | 8 |
| 2.4 | Cyklus programu PLC [11] | 8 |
| 2.5 | CP-1000 [11] | 9 |
| 2.6 | Jazyky normy IEC 61 131 [9] | 12 |
| 2.7 | Vývojové prostředí Mosaic | 13 |
| 2.8 | Manažer projektu v prostředí Mosaic | 14 |
| 2.9 | Ovládání Mosaicu | 15 |
| 2.10 | Zjištění konfigurace sítě v PLC [11] | 16 |
| 2.11 | Nastavení sítě | 16 |
| 3.1 | Program v jazyce strukturovaného textu | 23 |
| 3.2 | Funkční blok rekuperace | 24 |
| 3.3 | Schéma propojení modulů zabezpečení domu | 27 |
| 3.4 | Schéma propojení systému foxtrot se senzorovou sítí | 28 |
| 3.5 | Formulář pro vstupní data ze senzorové sítě | 29 |
| 3.6 | Webové rozhraní Foxtrot | 29 |

Kapitola 1

Úvod

Již v minulém století lidé uvažovali o budoucnosti s domy, které se řídí samy. S domy, které mají vlastní inteligenci. S rozvojem technologií se již nyní začínáme částečně tomuto snu přibližovat. Nelze ještě dosáhnout u domů samostatného myšlení jak je prezentováno ve filmech. Lze však díky těmto systémům velmi zkvalitnit život a snížit náklady spojené s bydlením.

Pro svou bakalářskou práci jsem si zvolila systém Foxtrot firmy Teco a.s. Hlavním důvodem je komplexnost systému a jeho pořizovací cena, což je pro budoucí majitele těchto systémů zásadní kritérium. Dalším důvodem je i to, že se jedná o českého výrobce.

Cílem této práce bude nejen naprogramování inteligentního domu, ale i propojení tohoto systému s bezdrátovou senzorovou sítí. Samostatné programování a nastavení senzorové sítě není součástí mé bakalářské práce.

Kapitola 2

Analýza systémů

2.1 Inteligentní dům

2.1.1 Historie

První „elektornický dům“ byl postaven nadšenci v brzkých 60. letech minulého století. V této době ale technologie nebyla ještě na takové úrovni jako dnes, a proto se tyto domy nese-tkaly s přílišnou popularitou [3]. Pojem inteligentní budova 'smart house' byl poprvé ofici-álně použit v roce 1984 americkou Národní asociací stavitelů domů (American Association of House Builders). Ta spustila v roce 1985 projekt Smart-house. Tento projekt byl do-končen až v roce 1990 a v roce 1991 otevřen veřejnosti, kdy se setkal s velkým zájmem [7]. Za počátek zrodu jednotné koncepce inteligentní elektroinstalace je považován rok 1987, kdy firmy Berker, Gira, Metena a Siemens založily společnost Insabus Gemeinschaft. Cílem této společnosti bylo vyvinout systém pro měření a regulaci budov. Vzhledem k velkému zájmu o tyto technologie se v roce 1990 změnila firma na nadnárodní nezávislou organi-zaci EIBA (European Installation Bus Association), jejímž hlavním cílem bylo vytvoření jednotného standardu technologií od různých výrobců, aby jednotlivé komponenty systému spolu bezchybně spolupracovaly [5].

V České republice lze za „zrození“ inteligentních domů považovat projekt německého architekta Ludwig Meis van der Rohe, výstavbu vily Tugenhat, která se nachází v Brně. Ve vile je implementováno vytápění, elektronicky ovládaná okna a zabezpečení pomocí fotobuněk [7].

2.2 Co je to inteligentní dům ?

Myšlenka inteligentního domu vznikla již v padesátých letech minulého století. Tento termín je možné interpretovat jako dům, který je vybavený počítačovou a komunikační elektroni-kou, která vzájemnou interakcí reaguje na potřeby obyvatel domu, za účelem zvýšit komfort užívání domu, snížení spotřeby energií a zvýšení bezpečnosti domu. Toto pojetí je ovšem velmi rozsáhlé. V dnešní době se pojem inteligentní dům používá ve velkém spektru mož-ností od domu vybaveného bezpečnostním kamerovým systémem, až po plně automati-zované budovy [13]. Z tohoto důvodu definoval Richard Harper (autor publikace Inside the Smart Home) pět kategorií inteligentních budov v závislosti na výši inteligence. Každá kategorie pak rozšiřuje kategorii předešlou [3].

1. Obsahující inteligentní zařízení a systémy

V domě jsou implementována zařízení a systémy pracující samostatně bez vzájemné interakce. Jako příklad může být uveden právě kamerový bezpečnostní systém, nebo například ovládání světel pomocí fotobuněk.

2. Obsahující inteligentní komunikující zařízení a systémy

Systémy implementované do budovy vzájemně komunikují, čímž zdokonalují svou činnost. Například po uzamčení domu pomocí bezpečnostního kódu se zapnou kamery a vypnou se některá elektronická zařízení.

3. Propojený dům

Dům obsahuje kromě vnitřní komunikační sítě i vnější, která slouží pro vzdálené ovládání systému a přístupu k vnějším informacím. Například je možné na obrazovce implementované v domě zobrazovat aktuální předpověď počasí, kterou si systém může stáhnout z některého externího serveru s těmito údaji.

4. Učící se dům

Systém je schopný předvídat pomocí nashromážděných dat potřeby uživatele a pomocí toho řídit optimálně jednotlivé akce. Výhodou tohoto řešení je snížení nákladů na programování domu, jelikož program si dům vytváří sám.

5. Pozorný dům

Na rozdíl od učícího se domu, pozorný dům pracuje s aktuálními daty, tedy aktuální situací v domě, kterou vyhodnocuje a na základě tohoto rozhoduje o akcích v domě.

2.3 V současnosti používané systémy

S rozvojem elektrotechniky se na trhu začínají objevovat různé systémy, zajišťující inteligentní domácnosti. Všechny tyto systémy mají společné jádro komponent které umí ovládat. Velmi se ale liší možnostmi modifikace, propojení a kategorií inteligence zařízení.

- **Loxone** je systém velmi podobný systému Foxtrot. Je řízený centrální PLC jednotkou propojenou s periferiemi. Existuje i v bezdrátové verzi, což umožňuje instalaci i do již hotových budov. Na rozdíl od Foxtrotu systém Loxone nemá v nabídce tak velké množství periférií, tento nedostatek lze ale úspěšně nahradit součástmi z jiných systémů. Hlavní problém ovšem nastává v implementaci, kdy software na vytváření systémové logiky je pouze parametrický. To znamená, že jediné co je možné udělat je pospojování již předvytvořených funkčních bloků a jejich následné nastavení. Tento problém velmi omezuje možnosti uživatele v nastavení případných specifických požadavků, kdy by byla nutná vytvoření softwaru pro dům firmou Loxone.
- **iNELS** systém podporuje dnes převažující chápání inteligentní domácnosti. Nabízí jednotlivé prvky, které spolu vzájemně nejsou propojeny centrální jednotkou. Tyto prvky jsou samostatně ovládané pomocí mobilní/počítačové aplikace. Systém je spíše vhodný do již postavených budov, jelikož není potřeba žádných stavebních úprav. Vnímala bych ho ale spíše jako zajímavý doplněk než jako systém pro regulaci domu.

- **Fibaro** je velmi podobný systém jako iNels, ovšem jejich produkty jsou přímo určeny k oživení domu. Prvky jsou opět ovládány bezdrátově pomocí mobilní aplikace, nebo pomocí televize.
- **Control4** tento systém má již řídicí jednotku. Rozdíl mezi Control4 a Foxtrotem je ovšem v tom, že tato jednotka není určena na DIN lištu. Systém je tedy koncipován tak, že jednotka se nachází v místnosti a je propojená s dalšími prvky v ní. Systém je tedy vytvořen spíše jako domácí multikino než jako řídicí systém.
- **xComfort** je velmi rozsáhlý systém s množstvím periferních modulů. Většina komunikuje bezdrátově a systém je řízen centrální jednotkou. Rozdíl je opět v umístění jednotky – v tomto případě se nachází přímo na zdi jedné z místností, protože obsahuje i displej. U tohoto systému jsou opět pouze aplikace pro konfiguraci, samostatný vývoj tak není možný.
- **GILD** rozsahově i principiálně velmi podobný Foxtrotu, obsahuje všechny potřebné periferní moduly pro regulaci domu. Narozdíl od předchozích systémů ale nemá konfigurační software založený na funkčních blocích, ale na obrázcích. Tento fakt je velmi vhodný pro simulaci, ovšem vývoj systému to tím velmi degraduje

2.4 Základní principy inteligentní elektroinstalace

Inteligentní elektroinstalace je založena na vzájemné komunikaci jednotlivých prvků pomocí datové sběrnice. To na rozdíl od klasické elektroinstalace, kde jsou všechny prvky napojené na silový zdroj, což umožňuje pouze stavy vypnuto a zapnuto, dovoluje vyšší formu komunikace.

2.4.1 Prvky inteligentního systému

Aktor Jedná se o aktivní prvky systému, které vykonávají (většinou na základě údajů od některého senzoru) akci. Jsou to například spínače motorů (žaluzií), termostatické hlavice nebo alarmy.

Senzor Pasivní prvky systému, které slouží především ke snímání dat. Informace pak předávají po sběrnici. Příkladem jsou spínače, teploměry, PIR čidla[6].

V dnešní době jsou většinou v jednotlivých zařízeních systému implementovány jak senzory, tak aktory. V takovém případě můžeme toto zařízení nazvat multifunkčním [1].

2.4.2 Rozdělení systémů

Za základní rozdíl mezi jednotlivými inteligentními systémy lze považovat zda se jedná o systém centralizovaný, nebo decentralizovaný. Toto je ovšem pouze základní rozdělení, jelikož oba tyto typy lze kombinovat, poté nám vzniká takzvaný hybridní systém.

Decentralizovaný systém Základním specifikem decentralizovaného systému je, že neobsahuje žádnou centrální jednotku, případně řídicí počítač. Jednotlivé prvky systému tak komunikují vzájemně. V tomto systému je tedy nutné, aby každé zařízení mělo vlastní procesor a bylo samostatně naprogramované.

Komunikace potom probíhá tak, že zařízení odešle na sběrnici určité informace. K těmto informacím má poté přístup každé zařízení a v závislosti na informaci poté některá zařízení vykonávají požadovanou akci.

Výhodou tohoto systému je variabilita, kdy změna funkčnosti lze provést výměnou jednoho prvku. Dále není systém závislý na funkčnosti centrální jednotky. To znamená, že při poruše jednoho prvku nedojde k výpadku celého systému.

Nevýhodou je cena jednotlivých prvků.



Obrázek 2.1: Decentralizovaný systém [2]

Centralizovaný systém Základním prvkem centralizovaného systému je centrální jednotka. Ta je přímo spojena se všemi ostatními prvky systému, se kterými vzájemně komunikuje.

Propojení je tedy přímo závislé na centrální jednotce, která přijímá data od jednotlivých prvků. Po vyhodnocení těchto dat, zasílá dalším prvkům informaci o tom, jakou akci je třeba provést, avšak jednotlivá zařízení spolu přímo nekomunikují.

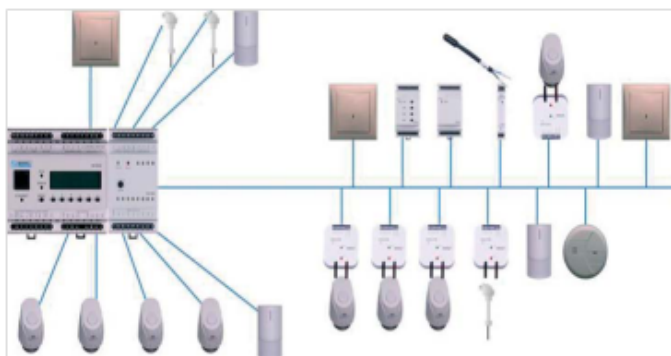
Hlavní výhodou tohoto systému je nízká cena jednotlivých prvků, jelikož nemusí obsahovat téměř žádnou inteligenci (veškeré výpočty a rozhodování provádí centrální jednotka).

Nevýhodou ovšem je úplná závislost na centrální jednotce, kdy při poruše dochází k úplnému výpadku systému.



Obrázek 2.2: Centralizovaný systém [2]

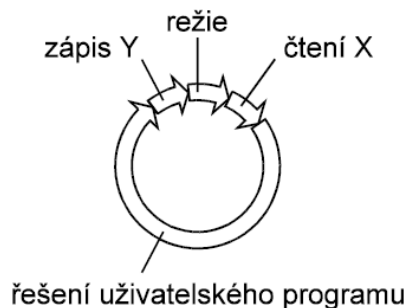
Hybridní systém Kombinací decentralizovaného a centralizovaného systému vzniká hybridní systém. Obsahuje tedy jak centrální jednotku, tak možnost připojení prvků na sběrnici, kdy sběrnice zprostředkovává komunikaci prvků s centrální jednotkou. [2]



Obrázek 2.3: Hybridní systém [2]

2.5 Tecomat Foxtrot

Programovatelný logický automat – PLC¹ je označení pro průmyslové počítače, využívané k automatizaci procesů v reálném čase. PLC tak prostřednictvím číslicových nebo analogových vstupů získává informace a řídí podle nich periferní zařízení. Hlavní rozdíl PLC oproti běžným počítačům je ve zpracování programu. PLC vykonávají program v cyklech. Řídící algoritmus je zapsán jako posloupnost instrukcí, které automat postupně vykonává. Po dokončení všech instrukcí jsou aktualizovány výstupní proměnné a vstupní stavy proměnných. Tento děj se nazývá jedním cyklem a neustále se opakuje viz. obrázek 2.4. [11]



Obrázek 2.4: Cyklus programu PLC [11]

PLC systémy lze rozdělit do dvou kategorií:

- **Kompaktní systémy** obsahují v jednom modulu CPU², vstupy/výstupy, základní podporu komunikace a zdroj. Jejich rozšiřitelnost je tak velmi omezená.
- **Modulární systémy** mají jednotlivé součásti rozděleny do modulů. Systém je pak složen z těchto modulů, díky čemuž je velmi snadné ho jakkoliv modifikovat. Zá-

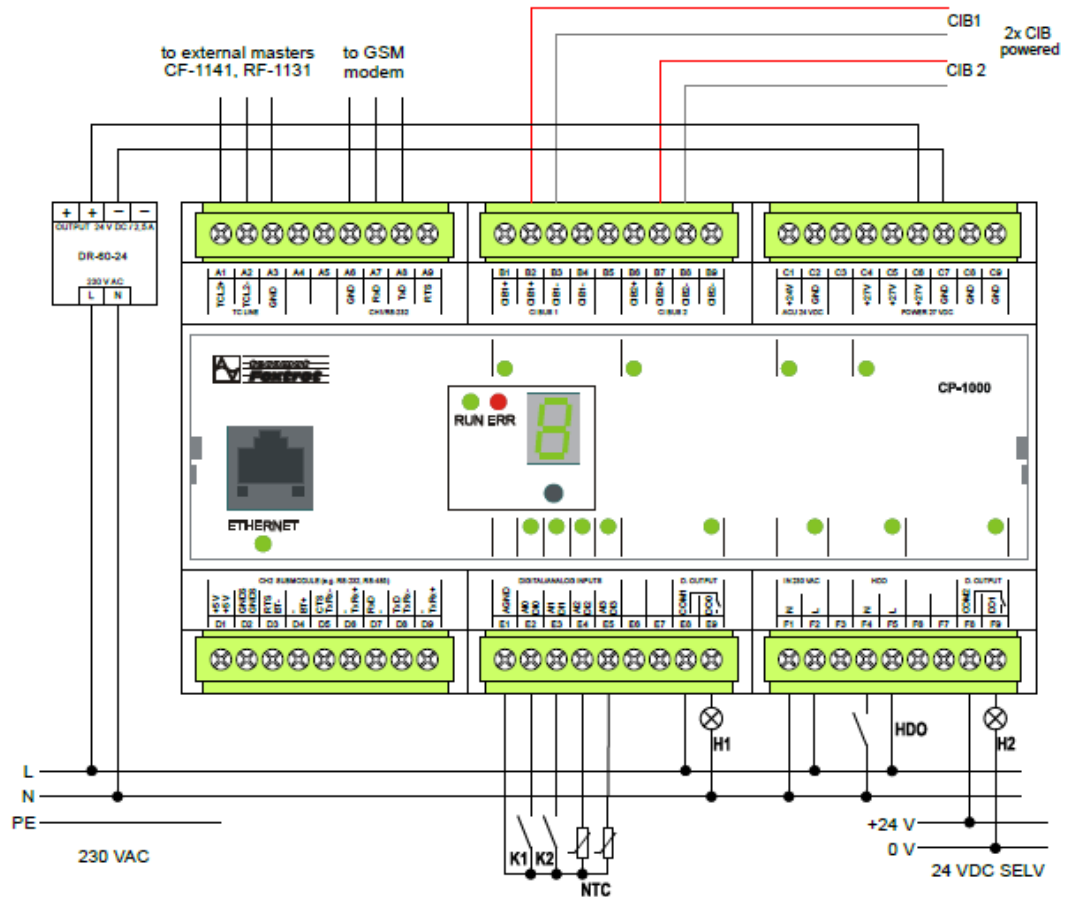
¹Programmable Logic Controller

²Central Procesor Unit

kladní moduly, které by takový systém vždy měl obsahovat je modul CPU, zdroje a vstup/výstupů.[15]

2.5.1 Foxtrot CP-1000

Tecomat Foxtrot je modulární řídicí systém a regulační systém od české společnosti Teco a.s.. Ve své bakalářské práci budu pracovat s centrálním jednotkou CP-1000, která se dá považovat za základní jednotku tohoto systému viz.obrázek 2.5. Základní parametry modulu CP-1000 jsou uvedeny v tabulce 2.1



Obrázek 2.5: CP-1000 [11]

³Common Installation Bus

| | |
|------------|---|
| Napájení | <ul style="list-style-type: none"> • 27 V DC, +10% , -15% |
| Příkon | <ul style="list-style-type: none"> • Max. 75 W |
| CPU | <ul style="list-style-type: none"> • Řada K |
| Paměť | <ul style="list-style-type: none"> • CMOS RAM – paměť pro uživatelský program, data a tabulky (192 + 64 KB) a DataBox(512 KB) • FLASH (EEPROM) – paměť pro zálohování uživatelského programu • slot pro MMC/SD/SDHC paměťovou kartu |
| Vstupy | <ul style="list-style-type: none"> • 4x víceúčelový vstup : analogový nebo binární • 2x binární vstup 230V AC • 2x samostatný reléový výstup |
| Komunikace | <ul style="list-style-type: none"> • 1x Ethernet 10/100 MB • 2x sériová linka(1x RS-232, 1x volitelná podle periferního modulu) • 2x CIB³sběrnice, pro připojení externích periférií • 1x TCL2 pro moduly rozšiřující počet I/O |
| Ostatní | <ul style="list-style-type: none"> • RTC obvod • Integrovaný web server |

Tabulka 2.1: Parametry modulu CP-1000

2.5.2 Norma IEC 61 131

Systém Foxtrot podporuje normu IEC 61 131. Tato norma definuje dvě základní části : společné prvky a programovací jazyky.

Společné prvky definované normou jsou :

- **Typy dat** jsou definované základní typy BOOL, BYTE, WORD, INT, REAL, DATE, TIME, STRING. Dále norma umožňuje definici vlastních typů, odvozených ze základních typů.
- **Proměnné** jsou v normě definovány jako lokální nebo globální. Hardwarové adresy pak mohou být k jednotlivým proměnným přiřazeny pouze v konfiguraci, zdroji, nebo programu, což umožňuje opakované využití na různých hardwarových platformách.
- **Konfigurace, zdroje a úlohy** : Konfigurace udává specifikace konkrétního systému, jako je například typ procesorových jednotek. V rámci konfigurace je pak možné definovat zdroje, tedy zařízení které jsou schopny vykonávat IEC programy. Uvnitř zdroje pak lze definovat úlohy, které řídí provádění určitých programů.
- **Programové organizační jednotky** je v normě IEC 61 131 jednotný název pro funkce, funkční bloky a programy.
- **Funkce** jsou definovány jak standardní (např. ADD pro sčítání), tak možnost uživatelských funkcí.
- **Funkční bloky** mohou být považovány za softwarovou abstrakci reálného hardwarového zařízení. Rozdíl mezi funkcí a funkčním blokem je v tom, že funkční bloky obsahují kromě algoritmů i data, takže mohou zachovávat informace z minulosti.
- **Program** kombinuje funkce a funkční bloky do jednotného systému.[9]

Programovací jazyky Ve standardu IEC 61 131 jsou definovány 2 typy jazyků – textové a grafické viz. obrázek 2.6.

- **Textové jazyky**

IL ⁴ – jazyk seznamu instrukcí. Jazyk se podobá jazyku symbolických instrukcí.

ST ⁵ – jazyk strukturovaného textu. Jazyk vychází z jazyků Ada, Pascal a C.

- **Grafické jazyky**

LD ⁶ – jazyk příčkového diagramu. Pochází z USA a představuje grafickou reprezentaci reléových obvodů.

FBD ⁷ – jazyk funkčního blokového schématu. Jazyk se velmi podobá elektronickým obvodovým diagramům, vyjadřuje chování jednotlivých funkčních bloků a jejich propojení. [9]

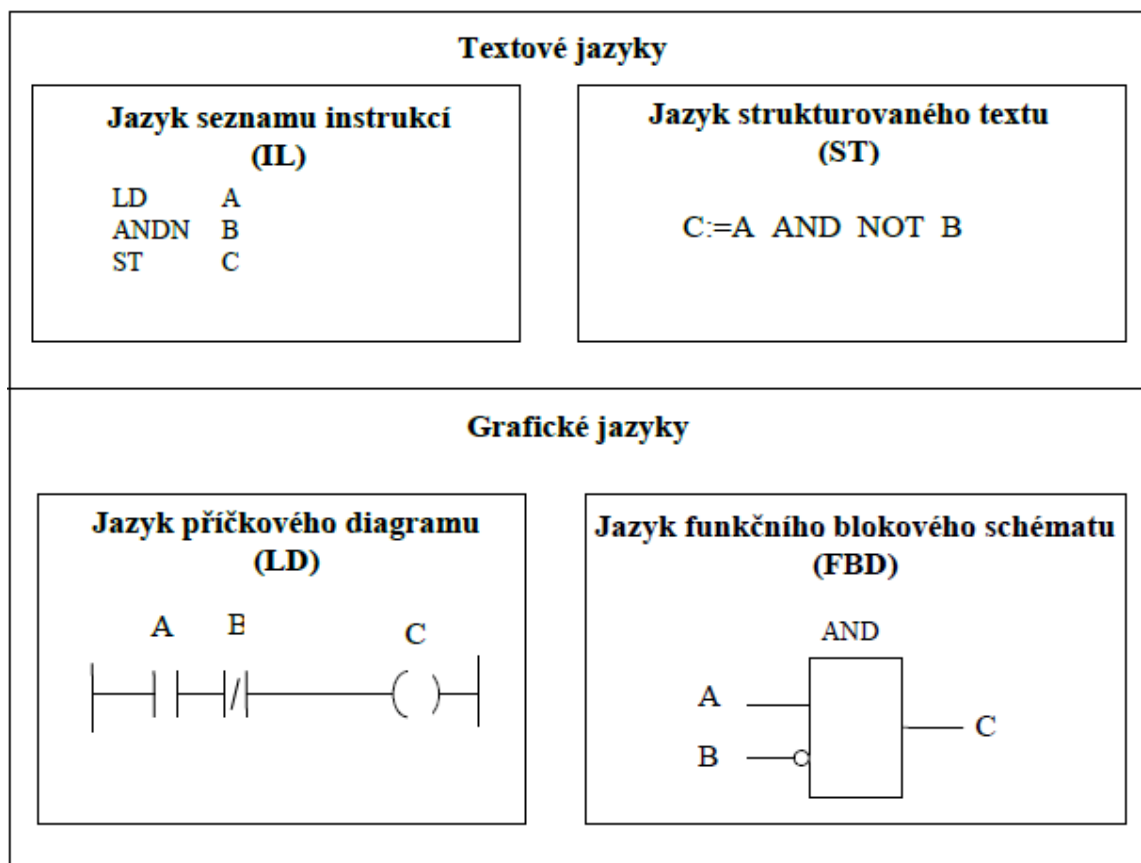
Při definici nového programu/funkce/funkčního bloku je možné zvolit si jazyk, ve kterém bude daný algoritmus zapsán. V jednom algoritmu není možné kombinovat více jazyků, je

⁴Instruction List

⁵Structured Text

⁶Ladder Diagram

⁷Function Block Diagram



Obrázek 2.6: Jazyky normy IEC 61 131 [9]

ale možné použít algoritmy definované v jiném jazyce. Například pokud hlavní program bude psaný v jazyce funkčního blokového schématu, je možné zde použít instanci která byla definována v jazyce strukturovaného textu. Tato možnost kombinování různých jazyků je velmi výhodná, protože pro různé části kódu může být vhodnější použití různých jazyků.

Ve své práci kombinuji jazyky ST a FBD. Pro psaní funkčních bloků a funkcí se mi jazyk strukturovaného textu jeví jako nejvhodnější, jelikož umožňuje psaní přehledného kódu pomocí instrukcí skoku a iterací. Oproti jazyku seznamu instrukcí je mnohem přehlednější a pro složitější algoritmy i vhodnější. Jazyk funkčního blokového schématu pak používám v hlavním programu pro spojení jednotlivých funkčních bloků, jednotlivá propojení jsou pak přehlednější.

2.5.3 Vývojové prostředí Mosaic

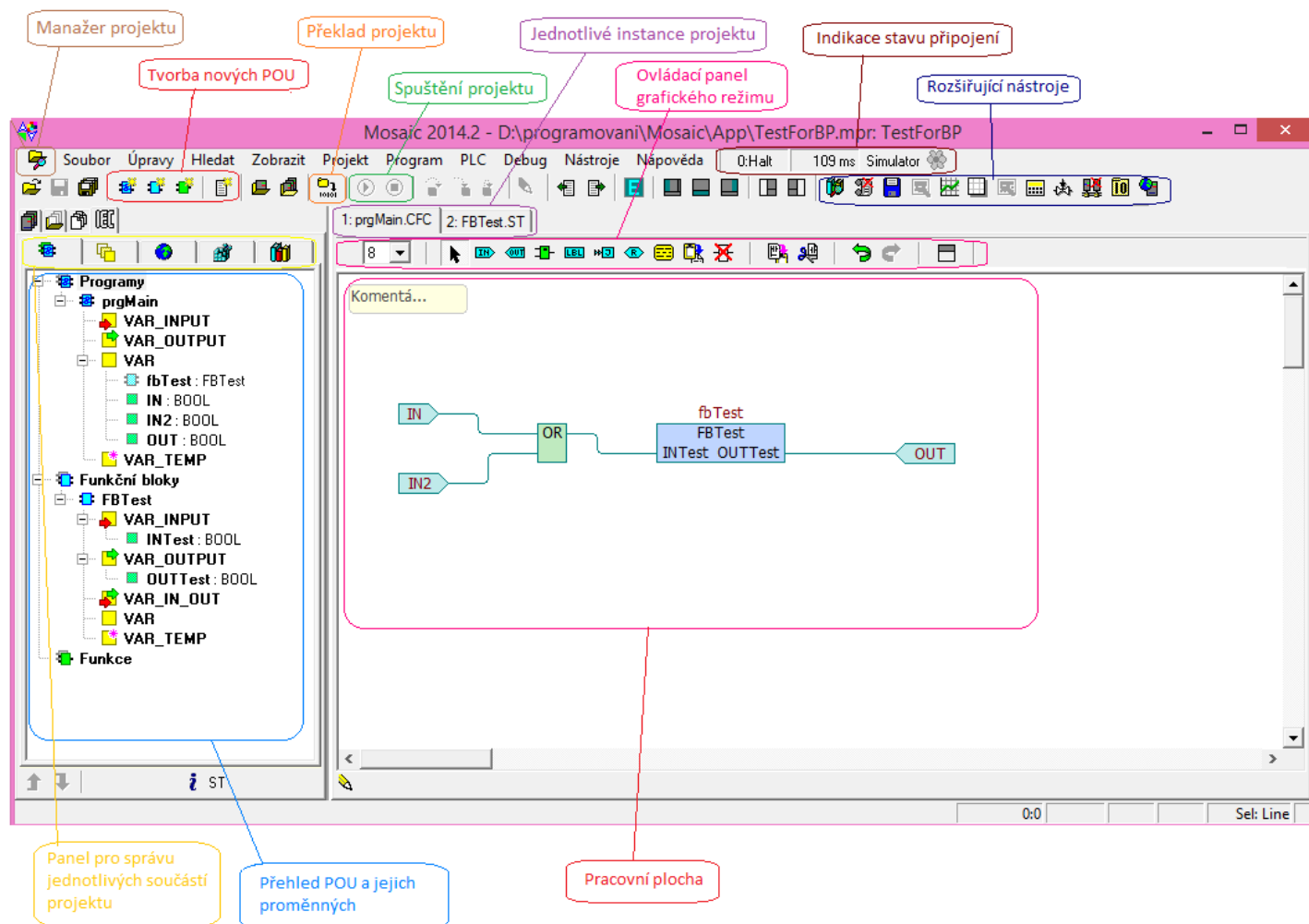
Mosaic je vývojové prostředí pro vývoj programů pro PLC systémy firmy Teco. Program tedy tvoří komplexní prostředí nejen pro nastavení PLC, tvorbu programů ale i tvorbu webového rozhraní, nebo vytváření větších sítí PLC jednotek.

Základní ovládání programu Mosaic viz. obrázek 2.7

- **Manažer projektu**

Jedná se o nástroj pro správu a konfiguraci systému (obrázek 2.8). Má tři hlavní části

- Nastavení připojení, kde je možné se připojit buď přímo k danému PLC, nebo



Obrázek 2.7: Vývojové prostředí Mosaic

pro účely ladění programu k simulaci PLC.

- Nastavení a konfigurace hardware, kde je možné nastavit periferní zařízení, komunikační kanály a i samotný typ PLC.
- Nastavení softwaru, pro verzování a tvorbu vlastních knihoven.

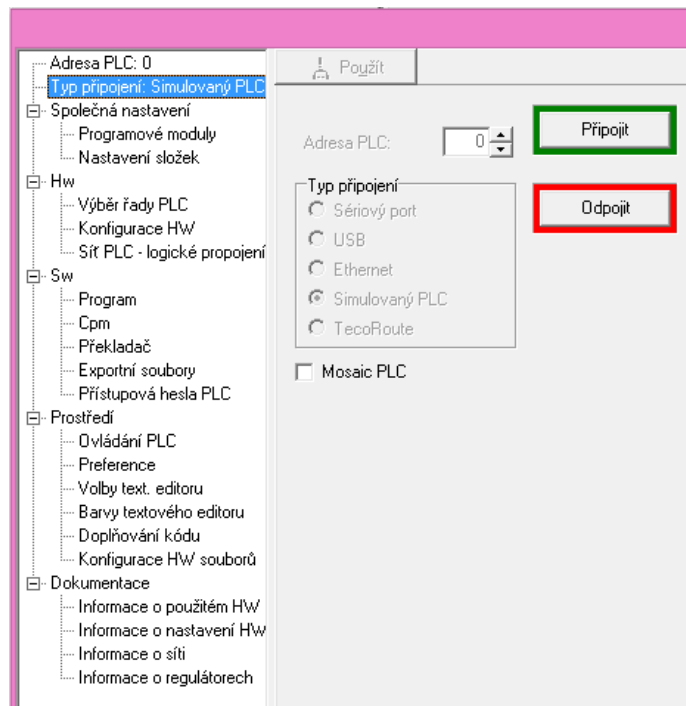
• Tvorba nových POU

Pomocí tlačítek lze vybrat typ nové POU ⁸. Je možné definovat v jakém jazyce má být psaná. Všechny POU vytvořené v projektu se zobrazí v přehledu jednotlivých instancí (obrázek 2.9a).

• Překlad projektu

Při překladu prostředí informuje o stavu překladu a o případných chybách (obrázek 2.9b)

⁸Programm Organisation Unit : Programová Organizační Jednotka



Obrázek 2.8: Manažer projektu v prostředí Mosaic

- **Spuštění projektu**

Zelená šipka indikuje nejen spuštění projektu, ale i nahrání algoritmu do paměti PLC. Proto je nutné nejprve nastavit v manažeru projektu s jakým PLC a jakou cestou budeme komunikovat. Zastavení pak znamená uvedení PLC do režimu HALT, tedy celé PLC přestane pracovat. Proto je lepší v případě ladění programu k vypnutí použít symbol tužky, která vypne pouze simulaci v PC.

- **Jednotlivé instance projektu**

V záložkách jsou zobrazeny námi otevřené části projektu na kterých právě pracujeme a je možné mezi nimi přepínat.

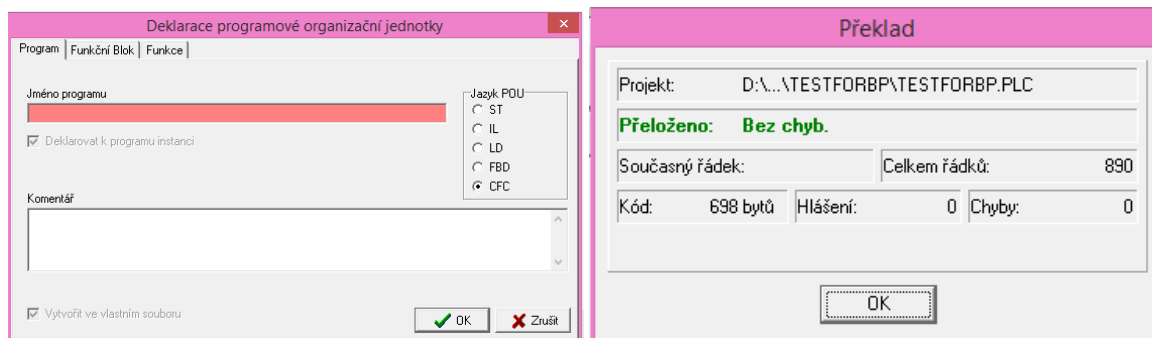
- **Ovládací panel grafického režimu**

Při psaní programu v grafických jazycích jsou zde zobrazeny prvky, které je možné do programu vkládat. Grafické jazyky, slouží hlavně lidem pro zjednodušení psaní programu. Mosaic je sám převádí do jazyka strukturovaného textu, který je možné zobrazit a editovat v případě potřeby.

Význam tlačítek zprava : zvětšení písma; editace pomocí myši; vložení vstupní proměnné; vložení výstupní proměnné; vložení POU; vložení návěští; vložení skoku; vložení RETURN; vložení komentáře; vložení instance ze schránky(CTRL + V); zrušení instance; kopírovat do schránky (CTRL + C); přesunout do schránky(CTRL + X); zpět; vpřed; zobrazení strukturovaného textu.

- **Indikace stavu připojení**

Informuje o režimu, ve kterém se právě PLC nachází (HALT, RUN), o typu připojení (ETHERNET, simulované PLC) a udává bližší specifikace spojení (UDP, IP adresa).



(a) Tvorba nových POU

(b) Překlad projektu

Obrázek 2.9: Ovládání Mosaicu

- **Rozšiřující nástroje**

Nástroje pro lepší a kvalitnější zpracování pro uživatele.

- **Průzkumník knihoven**, zajišťuje připojování knihoven k projektu, které mohou výrazně snížit časové náklady na tvorbu projektu. Uživatel sám může vytvářet vlastní knihovny a tím zajišťovat možnost znovupoužitelnosti jednotlivých prvků.
- **DataLogger** je nástroj pro ukládání dat ze systému, pro možnost následné analýzy. Tato data jsou ukládána na paměťovou kartu, proto je nutné aby jí bylo PLC vybaveno [10].
- **GraphMaker** slouží ke grafickému zobrazení průběhu daných proměnných a jejich následné analýze [8].
- **Nastavení V/V** je funkce pro nastavení periférií, dostupná i z manažera projektů.
- **WebMaker** slouží k vytvoření webového uživatelského rozhraní, přímo propojeného s jednotlivými proměnnými ke vzdálenému ovládání systému.

- **Panel pro správu jednotlivých součástí projektu**

Jednotlivé záložky umožňují modifikovat samostatné části projektu.

Význam tlačítek zprava : POU a jejich proměnné; Datové typy definované v projektu; Globální proměnné; Konfigurace; Knihovny.

- **Přehled POU a jejich proměnných**

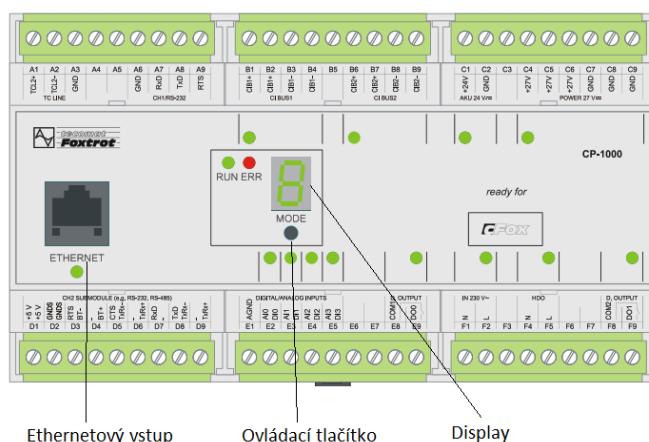
Přehledné zobrazení všech POU definovaných v projektu a jejich proměnných. Je zde možnost přímé modifikace instancí.

- **Pracovní plocha**

2.5.4 Prvotní nastavení

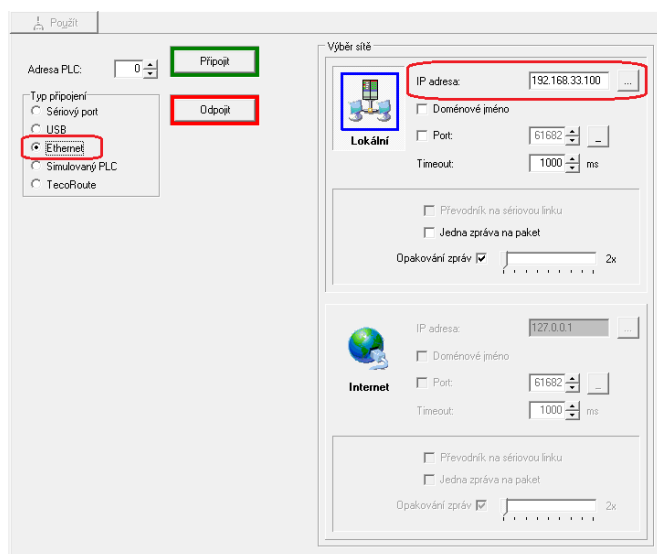
Vzhledem k tomu, že nejjednodušší a nejefektivnější propojení PC s centrální jednotkou je pomocí ethernetového kabelu, je nutné udělat prvotní nastavení systému a připojení k PLC. Po tomto nastavení bude možné již do jednotky nahrávat program. Každá centrální jednotka má v sobě přednastavenou IP adresu, masku sítě a bránu. Tyto informace je možné přičíst

přímo z jednotky po jejím zapojení. Po zmáčknutí a držení černého tlačítka umístěného na přední straně PLC se postupně na displeji tyto informace zobrazí (Obrázek 2.10).

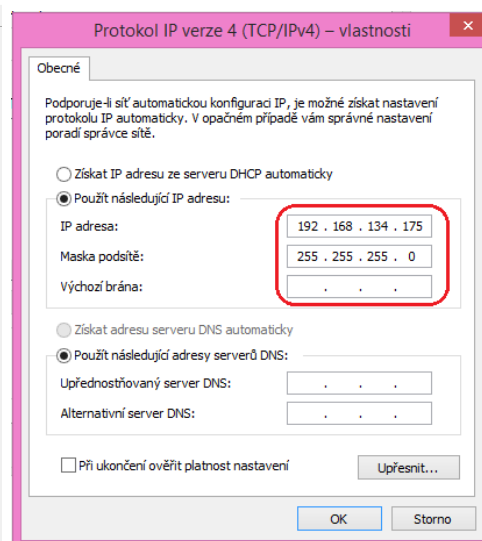


Obrázek 2.10: Zjištění konfigurace sítě v PLC [11]

V manažeru projektu, sekce typ připojení, vybereme Ethernet a nastavíme načtenou IP⁹ adresu (obrázek 2.11a). Zároveň je nutné nastavit v počítači síť ve stejném rozsahu, jako je síť nastavená v PLC – v případě přímého propojení PLC s počítačem musí být nastavena LAN síť (obrázek 2.11b).



(a) Nastavení připojení v Mosaic



(b) Nastavení sítě v PC

Obrázek 2.11: Nastavení sítě

⁹Internet Protocol

2.6 Bezdrátové senzorové sítě

Bezdrátová senzorová síť je tvořena autonomními, vzájemně komunikujícími senzory a vstup-výstupním rozhraním pro vnější komunikaci. Tyto senzory mohou snímat některé z fyzikálních veličin, jako je například teplota, světlo nebo vlhkost.[12]

První senzorové sítě byly tvořeny pro vojenské účely. Za první senzorovou síť lze považovat systém SOSUS, který vyvíjela americká armáda v 50. letech minulého století. Systém byl určen pro monitoring Atlantského oceánu¹⁰. [4]

V dnešní době našly bezdrátové senzorové sítě uplatnění kromě vojenství i v dalších oborech. Například v přírodních vědách, kde je možné těmito senzory pokrýt za nízké náklady rozlehlé oblasti. Toto je využíváno k neinvazivnímu sledování dané lokality, což může být použito k vědeckým¹¹, nebo zemědělským¹² účelům.

Dalšími významnými obory jsou průmysl a stavebnictví, kde jsou senzorové sítě využívány především ke sběru dat, pro automatizované ovládání.

2.6.1 Základní principy bezdrátových senzorových sítí

Bezdrátová senzorová síť se skládá z jednotlivých senzorových uzlů¹³ a vstup-výstupní brány¹⁴. Pro implementaci bude použit hardware firmy Crossbow, konkrétně produktová řada IRIS. Tato řada sestává z senzorových uzlů¹⁵ a základní stanice¹⁶. Základní parametry jsou uvedeny v tabulce 2.2.

Senzorový uzel Senzorový uzel je komponenta složená ze tří základních prvků – zdroje energie, komunikační jednotky a senzorů. Vzhledem k určení bývá zdroj energie většinou ve formě baterií. Z tohoto důvodu je nutné, aby jednotlivé uzly spotřebovávaly co nejméně energie, jelikož v některých případech může být výměna baterie i nemožná. Tohoto se dosahuje optimalizací komunikačních algoritmů.

Komunikační prvek se specializuje na propojení jednotlivých prvků sítě. Komunikace mezi jednotlivými uzly probíhá na základě uživatelského programu. Díky možnosti optimalizace směrovacích algoritmů je možné postavit velmi rozsáhlou síť, kdy ne všechny prvky na sebe musí vzájemně vidět.

Poslední částí senzorového uzlu je senzorová deska. Ta slouží především ke snímání veličin. Nasbíraná data jsou v závislosti na programu posílána zpět do základní stanice.

Základní stanice tvoří rozhraní mezi senzorovou sítí a počítačem/serverem. Posílá do senzorové sítě dotazy, na ty jí senzorová síť odpovídá. Přijatá data jsou pak pomocí USB konektoru přenesena do počítače, kde je možné je analyzovat.

2.6.2 Platforma JADE

JADE je JAVA platforma určená k vývoji softwarových agentů. Pro senzorovou síť použitou k propojení se systémem Foxtrot byl použit systém JAWS určený k řízení a monitoringu

¹⁰Systém se v dnešní době používá pro vědecké environmentální projekty

¹¹Např. výzkum flóry a fauny v chráněných lokalitách

¹²Např. sběr dat pro účely zefektivnění pěstování plodin

¹³Sensore nodes

¹⁴Base station

¹⁵Nodes

¹⁶Mote

| Základní stanice IRIS mote XM2110 | |
|--|--|
| Procesor | Atmel ATMega1281 (8MHz) |
| Rádiový transceiver | Atmel RF230 |
| Paměť | Sériová flash paměť (512kB) |
| | Flash paměť pro program (128kB) |
| | RAM paměť (8kB) |
| | EEPROM konfigurační paměť (4kB) |
| napájení | 2 AA baterie (2.7 - 3V) |
| uživatelské rozhraní | 2x LED dioda |
| Ostatní | 10-bitový ADC převodník |
| | 51 pinový slot pro připojení senzorové desky |
| Senzorová deska MTS400 | |
| Měrné veličiny | Teplota |
| | Barometrický tlak |
| | Osvět |
| | Vlhkost |

Tabulka 2.2: Základní parametry řady IRIS [4]

senzorové sítě. JAWS se skládá ze tří agentů, kteří řídí senzorovou síť. Propojení platformy se sítí je pomocí BSCOM rozhraní.

Agenti:

- GateKeeper obsahuje HTTP¹⁷ sever, který je schopný komunikovat BSCOMM rozhraním. Získané informace dále předává Service Providerovi.
- Service Provider slouží jako mezičlánek. Třídí přijatá data a na základě nich vyvolává příslušné aplikace.
- WSagent je podpora pro připojení senzorové sítě.

Dále je nutné mít program pro definici chování sítě a očekávaných výstupů. Pro potřeby inteligentního domu je program koncipován tak, že jednotlivé uzly komunikují přímo se základní stanicí. WSagent následně přijatá data uloží do předem definovaného souboru. Bližší znázornění aplikace viz. Schéma 3.4.

2.7 Raspberry Pi

Raspberry Pi je jednodeskový počítač vyvíjený britskou nadací Raspberry Pi Foundation. Počítač byl vyvíjen primárně pro zkvalitnění a rozšíření výuky informatiky, ale své uplatnění najde i v běžném využití [16]. Raspberry Pi je díky jeho komponentám možné považovat za plnohodnotný počítač a díky jeho ceně a velikosti je možné ho využít jako doplňkou periférii inteligentního systému. Dokáže totiž nahradit některé z originálních periférií, které ho ale vysoce převyšují cenou, a kromě toho i přidat funkcionalitu již nelze pomocí Foxtrotu dosáhnout, jako je například kamerový systém.

V mém projektu bude použito Raspberry Pi B+, jeho hardwarové parametry jsou uvedeny v tabulce 2.3. Jako operační systém je použit Raspbian, což je volně dostupná verze Linuxové distribuce Debian optimalizovaná pro hardware Raspberry Pi. Hlavním důvodem rozšíření systému o Raspberry je propojení Foxtrotu se zařízeními, které nejsou přímo kompatibilní s Foxtrotem. V mém případě se jedná o propojení se senzorovou sítí, která má jako výstup USB rozhraní jenž není Foxtrotem podporované.

| | |
|----------|---------------------------------------|
| CPU | ARM1176JZF-S (jednojádrový, 700MHz) |
| GPU | VideoCore IV, Multimedia Co-Processor |
| RAM | 512MB SDRAM |
| Úložiště | micro SD karta |
| USB | 4x USB 2.0 |
| Video | HDMI |
| Audio | 3.5mm jack, HDMI |
| LAN | Ethernet RJ45 10/100 |

Tabulka 2.3: Hardwarová specifikace Raspberry Pi [14]

¹⁷Hyper Text Transfer Protocol

Kapitola 3

Návrh realizace

Jako vzorový projekt jsem zvolila středně velký rodinný dům, který má 2 ložnice, obývací pokoj spojený s kuchyní, koupelnu, technickou místnost a předsíň. V domě bude systémem zajištěna regulace teploty a základní bezpečnostní systém.

Teplota v domě bude regulována pomocí podlahového topení, rekuperace a žaluzií. Logika vytápění/ochlazování pak bude závislá na venkovních teplotách a ročním období. Aktuální teplota bude snímána senzorovou sítí teplotních čidel, kde budou jednotlivé uzly umístěny v kritických bodech, kde je vyšší potenciál pro případné náhlé změny teploty.

Bezpečnostní systém se zaměří nejen na ochranu domu proti případnému vniknutí cizí osoby, ale i na bezpečnost vnitřního vybavení před případným poškozením. V rámci této bezpečnosti budou v domě umístěna záplavová čidla a regulační hlavice, které budou chránit dům před případným poškozením vodovodního systému a kouřová čidla pro případ požáru. Zabezpečení proti nechtěnému vniknutí cizí osoby je tvořeno venkovní sirénou a soustavou čidel ve dveřích a oknech. Celý bezpečnostní režim má dva režimy – denní a noční – pro možnost zabezpečení domu i v případě, že jsou obyvatelé doma.

3.1 Komponenty

Systém se bude skládat z těchto vzájemně propojených komponent.

- **CP-1000**

centrální PLC jednotka. Zajišťuje logiku systému a propojuje veškeré své periferie se zbývajících komponentami systému. K propojení s periferiemi je využita systémová sběrnice CIB1.

Periferie přímo připojené k PLC :

- **Napájecí zdroj (DR-100-24)**

Jeho hlavní úlohou je převod elektrické energie přiváděné ze sítě na proud o napětí 24V na kterém pracuje PLC.

- **Ovládání topení (C-OR-0008M)**

Podlahové vytápění je spínáno pomocí reléových výstupů. Modul obsahuje těchto výstupů 8, je tedy možné s tímto jedním modulem ovládat všechny topné jednotky v domě.

– **Ovládání rekuperace (C-HM-0308M)**

Obsahuje 6 relé, které je možné použít pro propojení systému s rekuperací. Pomocí těchto relé se budou zapínat jednotlivé stupně rekuperace a spouštět obtokový systém.

– **Žaluziový modul (C-OR-0202B)**

Modul se dvěma nezávislými relé, kterými lze ovládat dva různé spotřebiče. Pro potřebu ovládání žaluzií je ovšem nutné použít pro každou žaluzii jeden samostatný modul. Jedno relé je pak využito pro posun nahoru a druhé dolů.

– **Vodní ventil (CWX-15N)**

Vodní ventil obsahuje kulovitý uzávěr, který lze pomocí implementovaného servopohonu zavřít, nebo otevřít.

– **Kontaktní senzory (SA-201A)**

Senzory mají magnetické kontakty, které po rozpojení změní vysílaný signál. Tyto senzory jsou umístěny ve dveřích a oknech, ve spodní části – díky tomu nevadí pokud jsou otevřena na ventilaci. Jejich vývod je veden do modulů pro ovládání žaluzií, čímž se propojí se systémem.

Hlavní úlohou těchto senzorů je zajišťování bezpečnosti před cizími osobami. V mém projektu jsou navíc využity jako bezpečnostní prvek pro manipulaci se žaluziemi.

– **Pohybové čidlo (C-RQ-0600)**

Čidlo je umístěno v předsíni domu, kde zdvojuje bezpečnost proti vniknutí.

– **Čidlo osvitů (C-RI-0401S)**

Snímání venkovního osvitů, k vylepšení regulace vnitřní teploty. Slouží i k automatizovanému ovládání žaluzií.

– **Požární čidlo (SD-282ST)**

Jedná se o kombinovaný optický detektor kouře a detektor vysoké teploty. V zařízení je implementována i siréna.

– **Záplavový senzor**

– **Tlačítka pro ovládání žaluzií (C-WS-0200R + C-WS-0400R)**

Pro ovládání žaluzií v obývacím pokoji je použita čtyř-tlačítková verze, v pokojích jsou pak pouze dvou-tlačítkové.

– **Klávesnice (C-WG-0503S + ACM08E)**

Klávesnice je umístěná v předsíni domu. Slouží k uvedení bezpečnostního systému do aktivního režimu a zpět zadáním bezpečnostního kódu.

– **Venkovní siréna**

• **Podlahové vytápění**

Vytápění domu je realizováno pomocí elektrického podlahového vytápění. To spočívá v umístění topných kabelů pod povrch podlahy. Toto topení je pak zapínáno a vypínáno pomocí relé. U topení je nutné zajistit, aby se nepřehřálo. Z tohoto důvodu bude spouštěno v krátkých časových intervalech, kdy střídavě hřeje a chladí.

• **Rekuperační jednotka**

rekuperační jednotka zajišťuje rozvod čerstvého vzduchu v domě. Princip rekuperace spočívá v tom, že venkovní studený vzduch ohřívá pomocí vnitřního teplého vzduchu,

čímž zajišťuje menší tepelné ztráty, než běžné větrání. Rekuperace může pracovat na tři stupně, kdy každý vymění jiné množství vzduchu.

Pokud chceme, aby rekuperace vnitřní vzduch ochladila, je nutné, aby bylo možné zapnout obtok, tedy aby se vnější vzduch neohříval. Díky tomuto je při správné regulaci možno dosáhnout v domě i v letních měsících příjemné teploty bez nutnosti klimatizace

- **Raspberry Pi**

Základní úlohou Raspberry Pi v systému je propojení Foxtrotu se senzorovou sítí. Je ovšem možné stejným principem propojit s Foxtrotem i další nekompatibilní periferie.

- **Senzorová síť**

Jednotlivé uzly sítě jsou rozmístěny na strategických místech v domě. Informují pak systém o změnách teplot, díky nimž je pak dům regulován podle daných hodnot.

3.2 Implementace

Veškeré samostatné funkční bloky jsou propojeny v programu `main`, psaném v jazyce funkčního blokového schématu. Simulace vstupních hodnot je tvořena programem `simul`, v jazyce strukturovaného textu.

3.2.1 Příklad funkčního bloku

Funkční blok představuje abstrakci jednotlivých součástí systému. Hlavní rozdíl mezi funkčním blokem a funkcí je v jejich provádění. Funkce se provede pouze pokud je zavolána – vyhodnotí se na základě vstupů a předá výsledek. Funkcí blok se oproti tomu vyhodnocuje v každém cyklu a může mít více výstupů, které reprezentují výstupy dané součástky. Jako příklad implementace použijí funkční blok pro řízení rekuperace `fbRecuperation`.

Vnitřní strukturu funkčního bloku je vhodné psát v jazyce strukturovaného textu. Ten je rozdělen do několika částí. První částí jsou vstupní parametry definované v `VAR_INPUT` – `END_VAR`. Pokud by se jednalo o reálné zařízení, každý ze vstupů by byl jeden kabel připojený k dalšímu zařízení. Tyto parametry nelze ve funkčním bloku měnit. Co měnit lze jsou parametry výstupní v `VAR_OUTPUT`. Ty jsou v tomto bloku přímo napojeny na pomoci rozhraní na reálné vstupy periferního zařízení. V bloku `VAR` se nachází lokální pomocné proměnné. Ty mohou být i další funkční bloky, čímž lze simulovat i komplexnější zařízení. Tělo funkčního bloku obsahuje samotný kód funkce.

Při programování funkčního bloku je nutné aby si programátor uvědomoval cyklické provádění kódu. Jak je ukázáno na diagramu 2.4 program funguje tak, že nejprve načte aktuální hodnoty vstupů, následně vyhodnotí celý kód a až nakonec vloží na výstupy nové hodnoty. Na toto je potřeba pamatovat hlavně pokud má program vykonat změnu a posléze ji zrušit. V tomto případě musí být zrušení v kódu umístěno vždy před nastavením. Pokud by to tak nebylo, změna by se na výstupu vůbec neprojevila.

Jednotlivé funkční bloky se následně spojí v hlavním programu s danými vstupy a výstupy. Ty mohou být buď simulované programem, nebo reálné podle připojení jednotky k perifériím. V této části je vhodné použít jazyk funkčního blokového schématu. Ten je nejlépe znázorňuje jednotlivá zařízení a jejich vzájemné propojení, které připomíná reálné propojení zařízení pomocí elektronických spojů. Funkcí blok je zobrazen na obrázku 3.2. Pravá strana čtverce reprezentuje vstupy, levá strana výstupy.

```

VAR_INPUT
    varExchangerOff : BOOL := 0;
    varUnit : UINT := 0;
    varTurnOffTimeType : UINT := 0;
    varMode : UINT := 0;
END_VAR
VAR_OUTPUT
    varUnit1 : BOOL := 0;
    varUnit2 : BOOL := 0;
    varUnit3 : BOOL := 0;
    varExchangerOffOut : BOOL := 0;
END_VAR
VAR_IN_OUT
END_VAR
VAR
    fbTimerRecuperationON : TON;
    fbTimerRecuperationOFF : TON;
    tLongTimeOff : TIME := T#50m;
    tShortTimeOff : TIME := T#20m;
    tSuperLongTimeOff : TIME := T#10h;
    tOnTime : TIME := T#10s;
    varRecuperationRun : BOOL := 0;
    varRecuperationOff : BOOL := 0;
    tOffTime : TIME := T#0s;
    varRecuperationRunOK : BOOL := 0;
    varRecuperationOffOK : BOOL := 0;
END_VAR
VAR_TEMP
END_VAR

```

(a) Definice proměnných

```

if(varRecuperationRun)then
    varUnit1 := (varUnit = 1);
    varUnit2 := (varUnit = 2);
    varUnit3 := (varUnit = 3);
end_if;

2:
if(varRecuperationOffOK or
(not varRecuperationRun and not varRecuperationOff)) then
    varRecuperationRun := 1;
    varRecuperationOff := 0;
    varUnit1 := 1;
    varUnit2 := 0;
    varUnit3 := 0;
end_if;

if(varRecuperationRunOK) then
    varRecuperationOff := 1;
    varRecuperationRun := 0;
    varUnit1 := 0;
    varUnit2 := 0;
    varUnit3 := 0;
end_if;

if(varRecuperationRun)then
    varUnit1 := 1;
    varUnit2 := 0;
    varUnit3 := 0;
end_if;

3:
varUnit1 := (varUnit = 1);
varUnit2 := (varUnit = 2);
varUnit3 := (varUnit = 3);
varRecuperationOff := 0;
varRecuperationRun := 0;
end_case;

```

(b) Tělo programu

Obrázek 3.1: Program v jazyce strukturovaného textu

3.2.2 Topení

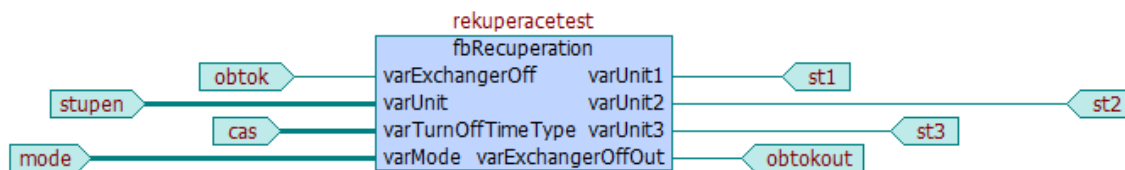
Funkční blok ovládající topení má dva vstupy: **varTurnOff** a **varEONOff** typu BOOL. **varTurnOff** slouží pro vypnutí topení, pokud je ve stavu 1, pak je topení vypnuto. Vstup **varEONOff** je určený pro vypnutí v době určené energetickým regulačním úřadem. Výstupem je **varHeatingON**, přímo ovládající daný výstup na ovládání topení.

Vnitřní realizace funkčního bloku pracuje se třemi časovači typu TON, kdy dva odměřují dobu pauzy topení – z důvodu nepřetížení tepelných vodičů v podlaze a jeden odměřuje dobu po kterou topení topí. Doby časovačů jsou pevně nastavené podle předchozího měření.

3.2.3 Žaluzie

Vstupy:

- **varControl**, vstup pro kontaktní senzor, který je umístěný u daných žaluzií. V případě, že by byl senzor rozepnutý – tedy měl hodnotu 0, pak se pohyb žaluzií vypne. Toto opatření bylo zavedeno kvůli bezpečnosti osob při automatickém spouštění žaluzií. Vyjímkou jsou jedny z žaluzií v obývacím pokoji, kde nejsou otevírací okna, proto je k nim přímo navedena proměnná jenž má pevně nastavenou hodnotu 1.
- **varUp** udává signál, že mají být žaluzie vysunuty nahoru. Žaluzie jedou pouze po dobu kdy má tato proměnná hodnotu 1.
- **varDown** udává signál, že mají být žaluzie spuštěny dolů. Žaluzie jedou pouze po dobu, kdy má tato proměnná hodnotu 1.



Obrázek 3.2: Funkční blok rekuperace

- **varHide** Narozdíl od ostatních vstupních proměnných je tato typu `BOOL R_EDGE`, což udává, že je aktivní při náběžné hraně. Po aktivaci, jsou žaluzie uvedeny do stavu kdy zastíní místnost před přímým sluncem – jsou tedy napůl zavřené.

V závislosti na aktuální poloze žaluzií při volbě `Hide` jsou žaluzie nejprve sesunuty úplně dolů a následně jsou pootočený do dané polohy.

Výstupy:

- **varDownOut** dává signál pro relé v žaluziovém modulu pro spuštění žaluzií.
- **varUpOut** dává signál pro vytažení žaluzií.

Žaluziový modul je opatřen zarážkou, kdy po dojetí do koncového stavu se motorek ovládající žaluzie vypne. I přesto je funkční blok opatřen vlastními prvky. Jedním z nich je nastavení času pohybu, kde pokud je tohoto času dosaženo, funkční blok přestane dávat signál k pohybu žaluzií. Dalším prvkem je koncová proměnná, která v případě vypršení celého nastaveného času uchovává stav žaluzií. Pokud tedy spustím žaluzie úplně dolů a následně se pokusím o stejnou akci, funkční blok zareaguje tak, že akci nevykoná.

Žaluziový modul je oproti ostatním modulům výjimečný v tom, že je možné ovládat jeho vstupy z více různých míst. V mém případě jde o ovládání pomocí tlačítek, z webového rozhraní a systémovými vstupy. Z tohoto důvodu je nutné celý modul „zabalit“ do dalšího bloku, který obsahuje logiku pro kompletaci těchto vstupů. Modul zajišťuje hlavně aby nedocházelo k nedefinovanému chování jako je například souběžné spuštění žaluzií nahoru i dolů.

Modul základní logiky žaluzií obsahuje vstupy nahoru, dolů a odstínění pro všechny lokace – tedy tlačítko, web a systém. Kromě těchto vstupů má navíc vstup i z osvitového čidla a časový údaj. Ty slouží k automatizovanému stažení žaluzií pokud venku nastane tma a k jejich následnému roztažení ráno v době určené uživatelem.

3.2.4 Rekuperace

Vstupy :

- **varExchangeOff** slouží k vypínání/zapínání výměníku. Pokud je ve stavu 1, pak se rekuperace přepne na obtok, tedy přestane ohřívat venkovní vzduch. V opačném případě jde nový vzduch přes výměník, kde je ohříván odváděným vnitřním vzduchem.

- **varUnit** proměnná typu **UINT**, která určuje na jaký výkonostní stupeň rekuperace poběží. Může nabývat hodnot 0-3, kde hodnota 0 udává vypnutou rekuperaci a ostatní hodnoty daný stupeň.
- **varTurnOffTimeType** Rekuperace funguje v několika režimech. V některých režimech se střídají doby, kdy je rekuperace zapnutá a kdy je vypnutá. Doba, kdy je v těchto režimech rekuperace zapnutá není závislá na aktuálním stavu domu. Ovšem doby kdy nepracuje na něm velmi závisí a tento vstup je určuje.

Může nabývat hodnot 0-2. Hodnota 0 udává, že dům není zamčený – tedy nacházejí se v něm jeho obyvatelé. Tento režim je vhodný hlavně pro denní dobu, kdy je nutné vzduch častěji obměňovat. Hodnota 1 slouží k nočnímu režimu, kdy není nutné vzduch obměňovat tak často vzhledem k nízké aktivitě obyvatel. Poslední stav je určen pro případ, že je dům zamčený. Předpoklad je takový, že v této době se v domácnosti nevyskytují lidé a díky tomu stačí vyměnit vzduch jednou za delší dobu.

- **varMode** určuje režim ve kterém se rekuperace aktuálně nachází. Nabývá hodnot 0-3, kde každá hodnota určuje jiný mód.

Pokud se proměnná nachází ve stavu 0, je rekuperace vypnutá. Tento režim je definován hlavně pro případ, že se chce některý z obyvatel koupat (při zapnuté rekuperaci velmi rychle proudí vzduch, což způsobuje pocit zimy). Mód 1 spouští rekuperaci v časových intervalech, podle proměnné **varTurnOffTimeType** a nastavuje stupeň výkonosti podle **varUnit**. Mód 2, se pak od tohoto velmi neliší. Pouze stupeň je vždy nastaven na 1, jde tedy o nejvíce automatický režim. Poslední mód pak slouží k úplnému ovládání uživatelem. Je tedy nastaven stupeň, který si zvolí a na tuto výkonnost pak běží neustále – nejsou zde časové cykly.

Výstupy :

- **varUnit1** zapnutí rekuperace na stupeň výkonu 1.
- **varUnit2** zapnutí rekuperace na stupeň výkonu 2.
- **varUnit3** zapnutí rekuperace na stupeň výkonu 3.
- **varExchangeOffOut** přepínání mezi obtokem a výměníkem.

3.2.5 Komplexní regulace teploty

Vstupy:

- **varSetTemperature** proměnná typu **REAL**, udává aktuální požadovanou teplotu nastavenou uživatelem. Jedná se pouze o jednu hodnotu, která je této logice dodávána z jiného funkčního bloku. Důvedem je možnost pro uživatele volit různé teploty pro různé části dne, týdne i roku. Funkční blok nastavující teplotu tak hlídá pomocí denního času a data jaká je aktuální požadovaná teplota. Tu předává řídicí logice.
- **varSenzorNetTemp** předává logice aktuální teploty v jednotlivých místnostech. Jedná se o pole hodnot **REAL**.
- **varHeatingSeason** **BOOL** hodnota udávající, zda je aktuálně topná sezóna, nebo není. Hodnota se nastavuje dle uživatelského nastavení.

- **varLight** je hodnota venkovního osvitu, podle ní systém rozpoznává noc a den. Hlavní využití je v ovládání žaluzií.
- **varMorningTime*** časový údaj zadný uživatelem, který udává v kolik hodin se mají v jednotlivých místnostech otevřít žaluzie.

Výstupy :

- Výstupy z bloku pro regulaci teploty odpovídají vstupům pro rekuperaci, žaluzie a podlahové topení ve všech místnostech domu.

Základním prvkem pro regulaci teploty je proměnná **varHeatingSeason**, která určuje zda je aktuálně topná sezóna, nebo není.

Topná sezóna by měla být v období zimy. V této době je teplota regulována hlavně pomocí topení. Pokud je v místnosti menší teplota, než je stanovená je nejdříve snaha o její vytopení pomocí obnovitelných zdrojů, což je v tomto případě sluneční energie. Proto jsou, pokud je venku světlo, otevřeny žaluzie. V závislosti na tepelných vlastnostech domu může být daná místnost vytopena jen díky této energii. Dalším krokem je spuštění podlahového vytápění.

I během topné sezóny ovšem může dojít k přetopení, v tom případě, pokud je venku světlo, se neprve stáhnou žaluzie do stínící polohy. Pokud tento proces nepomůže přepne se rekuperace na obtok, díky čemuž proudí do domu studený venkovní vzduch. Ochlazování venkovním vzduchem během topné sezóny není velmi časté a zástin pomocí žaluzií by měl většinu případů pokrýt.

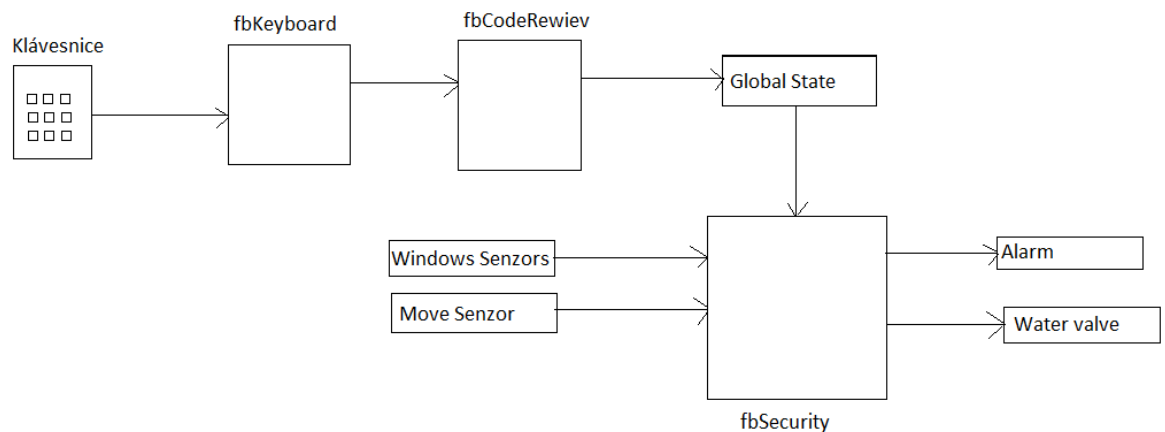
Mimo topnou sezónu je hlavní problém časté přetopení. Z tohoto důvodu, není vhodné zapínat topení. Pokud je teplota i přesto nižší než uživatelem definovaná, je rekuperační jednotka přepnuta z obtoku na výměnník¹ a jsou v případě denního světla otevřeny žaluzie.

Opačný případ je mnohem komplikovanější. Aby se zabránilo přívodu teplejšího vzduchu, než je uvnitř budovy je nutné aby přiváděný vzduch procházel přes výměnník a tím byl pomocí vnitřního vzduchu ochlazován. Vnitřní teplota se tím nesníží, ale nebude docházet k jejímu dalšímu nárůstu. Celkové ochlazení pak probíhá až v době, kdy je venkovní teplota nižší než vnitřní. To nastává převážně v nočních hodinách. Během tohoto času je rekuperace přepnuta na obtok čímž se vnitřní teplý vzduch vymění za studený venkovní. Po celou dobu by měly být žaluzie ve stínící poloze, aby nedocházelo k dalšímu oteplování domu.

3.2.6 Zabezpečení

Celkové zabezpečení domu je regulováno pomocí několika vzájemně propojených funkčních bloků (Obrázek 3.3). Je možno uvést dům do tří stavů – odemčeno, zamčeno a noční režim. Noční režim se od stavu zamčeno liší tím, že v případě přerušení čidel umístěných ve dveřích a oknech se nespustí venkovní siréna, ale pouze vnitřní bezpečnostní upozornění. Dále pak není hlášen pohyb ve vstupní místnosti domu.

¹Součástka rekuperace, kterou prochází nový vzduch a je ohříván vnitřním. Tím dojde k menším tepelným ztrátám.



Obrázek 3.3: Schéma propojení modulů zabezpečení domu

Klávesnice je fyzicky umístěna ve vstupní místnosti domu. Funkční blok klávesnice pak slouží ke snímání zadaného kódu a jeho přepisu do formátu řetězce. Důvod tohoto převodu je v tom, že z klávesnice převáděné znaky jsou přenášeny postupně a jako mocniny čísla 16. Funkční modul tato čísla převádí na číslice a ty ukládá do jednoho řetězce. Až po zmáčknutí odesílacího znaku je heslo vystaveno na výstup a předáno bloku pro kontrolu hesla.

Kontrola hesla spočívá v porovnání vloženého hesla s hesly pro zamčení a noční režim domu. Pokud zadané heslo odpovídá některému z daných hesel, přepne se stav a LED² dioda na klávesnici.

Modul zabezpečení Vstupy tohoto modulu jsou bezpečnostní čidla ve dveřích a oknech a pohybové čidlo. Stav ve kterém se dům nachází je uložen v globální proměnné, proto není nutné ho předávat jako vstupní parametr. Výstupem je pak ovládání venkovní sirény a vnitřního poplašného zařízení a ovládání vodních ventilů.

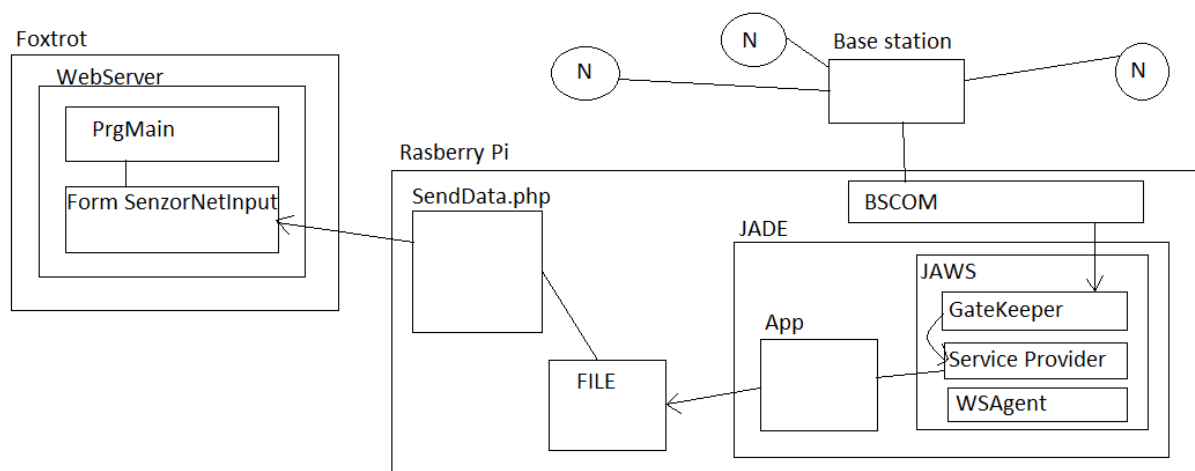
Pokud se pak dům nachází ve stavu zamčeno je nutné uzavřít vodní ventily, čímž je možné předejít možnému zaplavení domu v nepřítomnosti majitele. V případě že bude narušena bezpečnost domu je pak spuštěna velkovní siréna. Během nočního režimu dochází pouze ke kontrole zabezpečení a spouští se vnitřní poplašné zařízení. Tento režim je definován hlavně z důvodu možné přítomnosti dětí v domě. Pokud by pak nepozorovaně otevřely ve svém pokoji okno rodiče tento fakt zjistí a mohou předejít různým problémům.

3.3 Propojení systému se senzorovou sítí

Senzorová síť v projektu slouží pro snímání aktuální teploty v domě. Čidla jsou rozmístěna v domě ve všech místnostech, čímž mohou reagovat i na lokální změny teploty. Pokud se tak například v jedné místnosti sníží teplota začne se vytápět pouze tato místnost. Díky nezávislosti jednotlivých podlahových topení tak dochází ke konstantní teplotě ve všech místnostech a zároveň k úsporám elektrické energie. Jednotlivé uzly pak komunikují přímo se základní stanicí. Takto konstruovaná topologie díky menší náročnosti na přenos dat

²Light-Emitting Diode

šetří energii jednotlivých uzlů. Síť je propojená s obslužným programem pomocí BSCOM rozhraní.



Obrázek 3.4: Schéma propojení systému foxtrot se senzorovou sítí

Rozhraní mezi senzorovou sítí a systémem Foxtrot zajišťuje počítač Raspberry Pi. V něm je nainstalován obslužný program pro senzorovou síť na platformě JADE. Agenti systému JAWS zajišťují přenos dat na vstup aplikace. Po zpracování jsou hodnoty uloženy do souboru ve formátu desetinného čísla. Jako oddělovač je použit znak středníku.

Na straně Foxtrotu je pak implementován PHP³ skript `wnSendData`. Tento skript běží v nekonečné while smyčce. Skript načte data ze souboru, pokud se data od posledního čtení nezměnila do systému je neposílá. Při změně hodnot teplot skript vytvoří HTTP požadavek, ve kterém přiřadí jednotlivé hodnoty teplot k předem určeným polím formuláře. Následně data odešle. Každý cyklus je ukončen čekací dobou 60 sekund. Vzhledem k povaze dat není nutné aby byla odesílána častěji. Senzorová síť posílá nová data až po změně o určitý stupeň, čehož je možno dosáhnout až po delším časovém intervalu.

Data jsou tedy přenášena do formuláře umístěném na integrovaném webovém serveru systému foxtrot. Tento server umožňuje vytvořit několik stupňů oprávnění přístupu. Celý formulář má tedy nejvyšší stupeň ochrany díky čemuž se do něj uživatel domu nedostane. Toto je nutné bezpečnostní opatření, aby do formuláře mohl zapisovat pouze skript `wnSendData`.

3.4 Ovládání

Uživatelské rozhraní je vytvořeno pomocí integrovaného web serveru systému Foxtrot. Uživatelé jsou nastavena určitá přístupová práva, aby se zabránilo jeho přístupu do formuláře pro vstupy senzorové sítě. Po otevření stránky je načtena přihlašovací stránka. Tato stránka je zabezpečena šifrováním. Uživatel se přihlašuje pomocí uživatelského jména a hesla. Některým zařízením je možné přiřadit přístupová práva přímo pomocí jejich MAC⁴ adresy. V tomto případě není potřeba se přihlašovat a zařízení je přímo připojeno.

³PHP: Hypertext Preprocessor

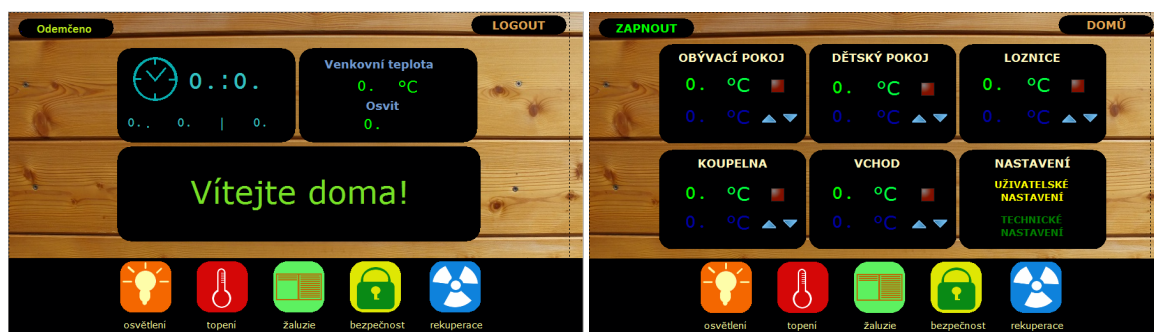
⁴Media Access Control

Vstupy ze senzorové site

| | | | |
|---------|--|--|--|
| Cidlo 1 | <input data-bbox="699 360 775 398" type="text" value="?"/> | Cidlo 4 | <input data-bbox="1043 353 1120 392" type="text" value="?"/> |
| Cidlo 2 | <input data-bbox="705 510 782 548" type="text" value="?"/> | Cidlo 5 | <input data-bbox="1050 504 1126 542" type="text" value="?"/> |
| Cidlo 3 | <input data-bbox="705 645 782 683" type="text" value="?"/> | <input data-bbox="892 631 1115 678" type="button" value="Submit"/> | |

Obrázek 3.5: Formulář pro vstupní data ze senzorové sítě

Webové stránky jsou rozděleny podle ovládání jednotlivých systémů. Díky možnosti přiřazení přístupových práv jednotlivým prvkům systému je možné vytvořit i řídicí části webových stránek pro možnost přenastavení některých systémových parametrů. V tomto případě jsou nastavena vyšší práva než u běžného uživatele.



(a) Úvodní stránka webového rozhraní

(b) Nastavení teploty

Obrázek 3.6: Webové rozhraní Foxtrot

Kapitola 4

Závěr

Požadavkem na tuto bakalářskou práci bylo vytvoření systému pro ovládání domu.

V první fázi práce jsem zjistila jakým způsobem fungují jednotlivé prvky použité v projektu. Následně bylo nutné vytvořit návrh jejich propojení a vzájemné komunikaci. Jako vzor budovy jsem zvolila rodinný dům o velikosti 2+kk a navrhla jsem pro něj základní model ovládacího systému. Tento model je zaměřený především na základní potřeby, což jsou regulace teploty a bezpečnost. Tento návrh lze rozvíjet o další prvky, jako je například ovládání světel, elektrospotřebičů, nebo přidáním dalších funkčních prvků typu klimatizace.

V této práci se mi podařilo propojit dva nekompatibilní systémy – Foxtrot a senzorovou síť – pomocí integrovaného webového rozhraní a počítače Raspberry Pi. Díky této formě rozhraní je možné systém Foxtrot doplnit o různé periférie, které nemusí být přímo k připojení k tomuto systému určeny. Příkladem dalšího rozvoje může být záložní zdroj, multimediální centra, nebo například bezpečnostní kamery.

Dalším krokem, který bych jako návrhář ovládacího systému pro rodinný dům volila by bylo externí webové uživatelské rozhraní. Především z důvodu možnosti šifrování pomocí HTTPS¹ formuláře², které u systému Foxtrot není u celých webových stránek možné a také z důvodu možnosti pořizování kamerového záznamu. Tuto možnost systém Foxtrot nemá, je jím možné snímat pouze statické fotografie.

¹Hyper Transfer Protocol Secure

²Důvodem je zajištění bezpečnosti dat i při přístupu z neověřených sítí například pomocí mobilního telefonu

Literatura

- [1] Brudková, I. M.; Veselý, I. P.: Inteligentní budovy. [online], [cit. 2014-10-19].
Dostupné z: <http://www.jilova.cz/Projekty/projekty-rozvoj-inteligentniBudovyStudium1.pdf>
- [2] bc. Dušan Hrušák: *Laboratorní úlohy pro model inteligentního domu*. [online],
Diplomová práce, České vysoké učení technické v Praze, Praha, 2012, [cit. 201-10-19].
Dostupné z: http://www.workswell.cz/wp-content/uploads/2012/06/DP_hrustak.pdf
- [3] Harper, R.: *Inside the Smart House*. London: Springer-Verlag, 2003, ISBN 1-85233-688-9.
- [4] Houšť, B. M.: *Monitorování stavu bezdrátových senzorových sítí agenty*. Diplomová práce, Vysoké učení technické v Brně, Brno, 2011.
- [5] inelin: Popis : Historie EIB. [online], 2013, [cit. 2014-10-16].
Dostupné z: <http://www.inelin.cz/popis/>
- [6] Kolář, M.: *Počítačové řízení prvků inteligentí elektroinstalace*. [online], Bakalářská práce, Vysoké učení technické Brno, Brno, 2008, [cit. 2014-10-16].
Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=9449
- [7] Kolinský, V.; Škodová, R. I.; Kubík, B. M.: Inteligentí dům není jen dálkové ovládání světel a rolet : Ja se v dálkovém ovládání uplatňují moderní komunikátory iPhone a iPad? [online], 10 2011, [cit. 2014-10-16].
Dostupné z: http://www.cmsys.cz/download/pdf/clanek_haustechnik_domintell.pdf
- [8] Teco a.s., Kolín: *Nástroj GraphMaker*. [online], první vydání, 2003, [cit. 2015-5-3].
Dostupné z: http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00327_01_Mosaic_GraphMaker_cz.pdf
- [9] Teco a.s., Kolín: *Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic*. [online], 10 vydání, 2007, [cit. 2015-5-3].
Dostupné z: http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00321_01_Mosaic_ProgIEC_cz.pdf
- [10] Teco a.s., Kolín: *Nástroj Datalogger*. [online], druhé vydání, 2011, [cit. 2015-5-3].
Dostupné z: http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00330_01_Mosaic_Datalogger_cz.pdf

- [11] Teco a.s., Kolín: *Programovatelné automaty tecomat forxtrot CP-1000, CP-1001, CP-1020*. [online], 6 vydání, 2014, [cit. 2015-5-3].
Dostupné z: http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00430_01_Foxtrot_CP_1000.pdf
- [12] Tyc, L.: *Aplikace bezdrátových senzorových sítí pro inteligentní domy*. [online],
Bakalářská práce, Vysoká škola polytechnická Jihlava, Jihlava, 2012, [cit. 2015-04-28].
Dostupné z: <https://is.vspj.cz/bp/get-bp/student/31037/thema/3158>
- [13] Valeš, M. M.: *Inteligentní dům*. Brno: ERA group spol. s r.o., 2006.
- [14] Voříšek, L.: Raspberry Pi v praxi podruhé: Detailní test miniaturního počítače za pár korun. [online], 3 2015, [cit 2015-5-7].
Dostupné z: <http://cdr.cz/clanek/raspberry-pi-2-recenze>
- [15] Wikipedia: Programovatelný logický automat. [online], 2014, [cit. 2015-5-3].
Dostupné z: http://cs.wikipedia.org/wiki/Programovateln%C3%BD_logick%C3%BD_automat
- [16] Wikipedia: Raspberry Pi. [online], 3 2015, [cit 2015-5-7].
Dostupné z: http://cs.wikipedia.org/wiki/Raspberry_Pi

Příloha A

Obsah CD

Příložené CD obsahuje následující soubory:

| | |
|----------------------------------|---|
| <code>intelligent_house</code> | adresář s projektem pro systém Foxtrot |
| <code>wnSendData.php</code> | php skript pro propojení systému se senzorovou sítí |
| <code>manual.txt</code> | textový soubor s návodem na spuštění projektu |
| <code>bachelor_thesis.pdf</code> | elektronická verze bakalářské práce |